Contents

Preface xiii
Book Layout xv
Acknowledgments xvii

1 Introduction to Real-Time Embedded Systems 1
1.1 Real-Time Embedded Systems 1
1.2 Example: Automobile Antilock Braking System 3
1.2.1 Slip Rate and Brake Force 3
1.2.2 ABS Components 4
1.2.2.1 Sensors 4
1.2.2.2 Valves and Pumps 5
1.2.2.3 Electrical Control Unit 7
1.2.3 ABS Control 8
1.3 Real-Time Embedded System Characteristics 10
1.3.1 System Structure 10
1.3.2 Real-Time Response 10
1.3.3 Highly Constrained Environments 11
1.3.4 Concurrency 12
1.3.5 Predictability 12
1.3.6 Safety and Reliability 13
1.4 Hard and Soft Real-Time Embedded Systems 13
Exercises 14
Suggestions for Reading 15
References 15

2 Hardware Components 17
2.1 Processors 17
2.1.1 Microprocessors 17
2.1.2 Microcontrollers 19
2.1.3 Application-Specific Integrated Circuits (ASICs) 19
2.1.4 Field-Programmable Gate Arrays (FPGAs) 19
2.1.5 Digital Signal Processors (DSPs) 20
2.1.6 Application-Specific Instruction Set Processors (ASIPs) 20
2.1.7 Multicore Processors 20
2.1.8 Von Neumann Architecture and Harvard Architecture 21
2.1.9 Complex Instruction Set Computing and Reduced Instruction Set Computing 22
2.2 Memory and Cache 23
2.2.1 Read-Only Memory (ROM) 23
2.2.2 Random-Access Memory (RAM) 24
2.2.3 Cache Memory 24
2.3 I/O Interfaces 26
2.4 Sensors and Actuators 27
2.5 Timers and Counters 29
Exercises 30
Suggestions for Reading 31
References 31

3 Real-Time Operating Systems 33
3.1 Main Functions of General-Purpose Operating Systems 33
3.1.1 Process Management 34
3.1.2 Memory Management 36
3.1.3 Interrupts Management 39
3.1.4 Multitasking 39
3.1.5 File System Management 39
3.1.6 I/O Management 41
3.2 Characteristics of RTOS Kernels 42
3.2.1 Clocks and Timers 42
3.2.2 Priority Scheduling 44
3.2.3 Intertask Communication and Resource Sharing 45
3.2.3.1 Real-Time Signals 45
3.2.3.2 Semaphores 46
3.2.3.3 Message Passing 46
3.2.3.4 Shared Memory 46
3.2.4 Asynchronous I/O 47
3.2.5 Memory Locking 47
3.3 RTOS Examples 48
3.3.1 LynxOS 48
3.3.2 OSE 49
3.3.3 QNX 49
3.3.4 VxWorks 49
3.3.5 Windows Embedded Compact 50
Exercises 50
4 Task Scheduling  53
4.1 Tasks  53
4.1.1 Task Specification  54
4.1.2 Task States  56
4.1.3 Precedence Constraints  58
4.1.4 Task Assignment and Scheduling  59
4.2 Clock-Driven Scheduling  59
4.2.1 Structured Clock-Driven Scheduling  62
4.2.1.1 Frames  62
4.2.1.2 Task Slicing  65
4.2.2 Scheduling Aperiodic Tasks  66
4.2.3 Scheduling Sporadic Tasks  68
4.3 Round-Robin Approach  69
4.4 Priority-Driven Scheduling Algorithms  70
4.4.1 Fixed-Priority Algorithms  70
4.4.1.1 Schedulability Test Based on Time Demand Analysis  72
4.4.1.2 Deadline-Monotonic Algorithm  76
4.4.2 Dynamic-Priority Algorithms  76
4.4.2.1 Earliest-Deadline-First (EDF) Algorithm  76
4.4.2.2 Optimality of EDF  78
4.4.3 Priority-Driven Scheduling of Aperiodic and Sporadic Tasks  82
4.4.3.1 Scheduling of Aperiodic Tasks  82
4.4.3.2 Scheduling of Sporadic Tasks  85
4.4.4 Practical Factors  85
4.4.4.1 Nonpreemptivity  85
4.4.4.2 Self-Suspension  86
4.4.4.3 Context Switches  87
4.4.4.4 Schedulability Test  87
4.5 Task Assignment  89
4.5.1 Bin-Packing Algorithms  89
4.5.1.1 First-Fit Algorithm  90
4.5.1.2 First-Fit Decreasing Algorithm  91
4.5.1.3 Rate-Monotonic First-Fit (RMFF) Algorithm  91
4.5.2 Assignment with Communication Cost  92
Exercises  94
Suggestions for Reading  97
References  97

5 Resource Sharing and Access Control  99
5.1 Resource Sharing  99
Contents

5.1.1 Resource Operation 100
5.1.2 Resource Requirement Specification 100
5.1.3 Priority Inversion and Deadlocks 101
5.1.4 Resource Access Control 103
5.2 Nonpreemptive Critical Section Protocol 103
5.3 Priority Inheritance Protocol 106
5.3.1 Rules of Priority Inheritance Protocol 106
5.3.2 Properties of Priority Inheritance Protocol 109
5.4 Priority Ceiling Protocol 111
5.4.1 Rules of Priority Ceiling Protocol 112
5.4.2 Properties of Priority Ceiling Protocol 114
5.4.3 Worst-Case Blocking Time 116
5.5 Stack-Sharing Priority Ceiling Protocol 119
5.5.1 Rules of Stack-Sharing Priority Ceiling Protocol 119
5.5.2 Properties of Stack-Sharing Priority Ceiling Protocol 121
Exercises 122
Suggestion for Reading 125
References 125

6 Concurrent Programming 127
6.1 Introduction 127
6.2 POSIX Threads 128
6.3 Synchronization Primitives 133
6.3.1 Race Conditions and Critical Sections 133
6.3.2 Mutex 134
6.3.3 Condition Variables 137
6.3.4 Semaphores 142
6.4 Communication among Tasks 148
6.4.1 Message Queues 149
6.4.2 Shared Memory 155
6.4.3 Shared Memory Protection 157
6.5 Real-Time Facilities 162
6.5.1 Real-Time Signals 162
6.5.1.1 Blocking Signals 163
6.5.1.2 Dealing with Signals 164
6.5.2 Timers 165
6.5.3 Implement Periodic Tasks 169
6.5.3.1 Using sleep() Function 169
6.5.3.2 Using Timers 172
6.5.4 Implement an Application with Multiple Periodic Tasks 173
Exercises 173
Suggestion for Reading 177
References 177